

WILSON ZHENG

(413) 429-6350 | wilsonz@andrew.cmu.edu | github.com/wilsonzheng0327

EDUCATION

Carnegie Mellon University (CMU)

Master of Science in Artificial Intelligence and Innovation, GPA: 4.1/4.3

Pittsburgh, PA

Aug 2025 – May 2027

The University of Virginia (UVA)

Bachelor's Degree in Computer Science & Physics, GPA: 3.91/4.0

Charlottesville, VA

Aug 2021 – May 2025

Echols Scholar, College Science Scholar, Dean's List

Relevant Courses: Computer Architecture, Compilers, Operating Systems, Cloud Computing, Machine Learning, Deep Learning, LLM Systems, AI Engineering, Generative AI for Music, NLP

WORK EXPERIENCE

Software Engineering Intern, Google, Sunnyvale, CA

May 2026 – Aug 2026

- Engineered a virtual fault-injection framework in C++20 for security controllers, simulating instruction-skip attacks to uncover secure-boot bypass vulnerabilities
- Authored hardened security patches mitigating glitch attacks via compiler-aware anti-glitch loop checks and reinforced verification primitives
- Debugged bare-metal firmware and analyzed raw Arm disassembly to confirm both vulnerabilities and their fixes
- Extended existing test suites to exhaustively evaluate Secure Boot verification logic under fault-injection scenarios

Software Development Engineer Intern, Amazon Web Services, Beijing, China

Jun 2025 – Aug 2025

- Engineered **Amazon Q**-powered AI unit test generation tool, raising code coverage to up to 90% across services
- Integrated into Amazon's **CI/CD** pipeline, enabling automatic test generation and saving ~10 hr/wk of manual testing
- Designed event-driven, scalable architecture on AWS to support asynchronous test generation workflows
- Developed multi-stage validation pipeline to detect and filter LLM hallucinations, significantly improving test reliability by eliminating invalid test cases before integration

FEATURED PROJECTS

SLang-Omni – Fun-ASR-Nano serving optimization contributor

June 2026 – Present

- Built a pre-LM audio encoder service hoisting full-context encoding ahead of LM admission with single-flight dedup and LRU embedding cache, 1.3× cold / 3.3× warm throughput at H100 concurrency 32
- Applied dynamic torch.compile to fuse the launch-bound 70-layer SANM encoder across variable-length LFR inputs, eliminated GIL-contention by stacking it on the standalone encoder thread, boosting throughput by an additional 11%
- Added a SenseVoice-style pad+mask batched encode kernel with SDPA, runtime-fused QKV, and FSMN v-reuse, ~2.3× faster encoder wall time and +56–84% throughput over the pre-LM service alone under sustained batching

NodeTorch – Open-source node-based visual ML model builder

Apr 2026 – Present

- Built an educational drag-and-drop node-based environment that lets users compose neural networks visually and train them, giving users a sandbox to experiment with model architectures without writing code from scratch
- Designed a layered, registry-driven execution engine separating graph runtime, type system, and ML domain, enabling new node types, datasets, and training paradigms to be added without modifying engine code
- Built rich interactive visualizations of every layer's behavior (forward activations, parameter distributions, attention patterns, loss surfaces) so learners can see why a model works, not just that it works
- Integrated an in-app AI assistant with a backend agent loop allowing any LLM to interact with the app through a schema-defined tool interface

Hypoll – Interactive Storytelling Social Platform

Oct 2025 – Apr 2026

- Built an AI-powered social platform using **React Native** and **FastAPI**, enabling users to create interactive audio content through conversations with AI host
- Architected end-to-end AI content pipeline: **GPT-4o mini** script generation, **Whisper** transcription, **Gemini** image generation and **ElevenLabs** TTS with word-level subtitle alignment
- Designed scalable backend with async **PostgreSQL** using **psycopg3** and deployed on **GCP**
- Implemented semantic search with **Qdrant** and OpenAI embeddings for content discovery and recommendation
- Engineered secure infrastructure (JWT, rate limiting, moderation) and stress-tested with **Locust** simulating hundreds of concurrent users

SKILLS

Programming Languages: Python, C++, C, C#, TypeScript, SQL, Bash, x86, CUDA

Frameworks and Tools: React, FastAPI, Redis, Qdrant, PyTorch, HuggingFace, OpenCV, GCP, AWS, PostgreSQL, Docker